Classifying Enterprise Architecture Analysis Approaches

Sabine Buckl, Florian Matthes and Christian M. Schweda

Chair for Software Engineering of Business Information Systems (sebis),
Technische Universität München,
Boltzmannstr. 3, 85748 Garching, Germany
{buckls,schweda}@in.tum.de
http://wwwmatthes.in.tum.de

Abstract. Enterprise architecture (EA) management forms a commonly accepted means to enhance the alignment of business and IT, and to support the managed evolution of the enterprise. One major challenge of EA management is to provide decision support by analyzing as-is states of the architecture as well as assessing planned future states. Thus, different kinds of analysis regarding the EA exist, each relying on certain conditions and demands for models, methods, and techniques.
In this paper we present a classification schema for EA analysis approaches to examine this topic. The classification schema is used to investigate the state-of-the-art of EA analysis by characterizing existing approaches according to the presented dimensions. Based on the results of this classification, future areas of research regarding EA analysis are derived.

Key words: Enterprise architecture management, EA analysis, ex ante/ex post analysis, behavioral/structural analysis, EA analysis classification

1 Introduction and motivation

In large enterprises, the application landscape, as the entirety of the employed business applications [1], is an important asset, which forms both a critical support factor for business and a costly investment constantly demanding maintenance operations. Consequently, managing the application landscape has not only recently become a challenge, today's enterprises have to address. The management of the application landscape is not an isolated task but a far-reaching endeavor, which has to be undertaken embedded in the context of enterprise architecture (EA) management. The holistic perspective of EA management sets up a framework for application landscape management, broadening the focus to related parts of the enterprise, such as business or infrastructure aspects (see Figure 1). These related parts of the overall architecture of the enterprise are themselves subject to other well-established management processes – the so called enterprise-level management processes, e.g. project portfolio management, strategies and goals management, and IT architecture management [2]. By linking these processes, to which application landscape management also belongs, to the EA management process, the overall alignment of business and IT in an enterprise can be fostered.

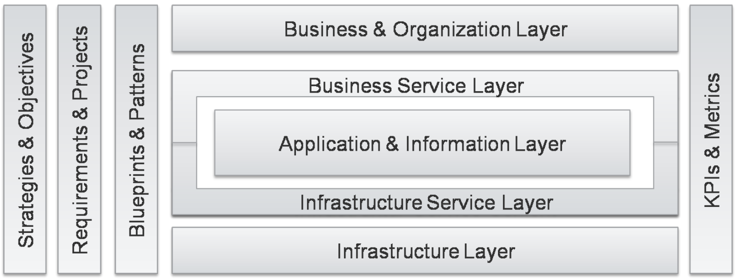

Fig. 1: Layers and crossfunctions of a holistic EA management approach [3]

In the light of the increased interest from practitioners, different approaches to EA management have been developed in academic research [4, 5, 6, 7], by practitioners [8, 9, 10, 11], standardization bodies [12, 13], and tool vendors [3]. These approaches provide frameworks, methods, and tools to facilitate EA management endeavors, but as of today they focus on *structural* aspects of the EA or of parts thereof, such as dependencies between business processes and business applications. Based on analyses of these aspects, decisions about the future of the EA are made and plans for a *managed evolution* (cf. [14]) are created. Currently, the dynamic behavior of the enterprise system[1] is not considered nor analyzed, although wrong decisions and inappropriate plans on the one hand might lead to high and long-term consequential costs, and on the other hand may have unforeseen impacts on the behavior of the highly dynamic system. The behavior is usually far more complex than the topological structure of the system is likely to indicate. The term *dynamic complexity*, which points to this fact, is a widely known and accepted in many management disciplines. First references date back to the 60^{ths} of the last century (cf. the *Forrester effect* [16]).

The dynamic complexity arising from the behavior of and interactions between the entities, which are considered to make up the EA, has yet not been an important topic in EA management research. This is especially surprising, as in some bordering areas many discussions on dynamic complexity have been undertaken, e.g. in the research of Ultra-Large-Scale Systems [17]. Even more prominent are the discussions in the area of business process management, where also techniques and methods for analyzing and evaluating dynamic complexity were developed (see e.g. [18, 19]).

Dynamic complexity in the enterprise system seems to us as a promising field of activity in EA management research, where the focus of the techniques for

[1] We use the term *enterprise system* to emphasize on the systems nature of an enterprise that is itself a system, which consists of systems, and is surrounded by systems [15].

EA analysis can be extended. As a guidance for further research, we develop a classification schema for EA analysis, which considers the different types of complexity but also aspects as e.g. the reach of the analyses. This schema is presented in Section 2. In subsequent Section 3, we classify existing approaches to EA analysis according to the dimensions of the schema. From there, we elaborate on the omissions in current research and shape different lines of activity to close the experienced gap. These ideas are presented in final Section 4.

2 A classification schema for EA analysis approaches

In the last years, quite a few scientific publications bearing the term *EA analysis* in their title have been published. Before discussing selected prominent approaches in the following, we should mention that albeit the increased popularity, no common definition of the term *EA analysis* has yet emerged. This may be caused by the plurality of techniques and methods that are subsumed under the term. Additionally, the approaches differ in respect to their coverage of the EA. To advance the development of this area, we do not try to give a comprehensive definition of the term, but present a classification schema (see Table 1) for analysis approaches. The dimensions of classification introduced therein, are explain below.

<table>
<tr><td>Body of Analysis</td><td colspan="2">structure</td><td colspan="2">behavior statistics</td><td colspan="2">dynamic behavior</td></tr>
<tr><td>Time Reference</td><td colspan="3">ex-post</td><td colspan="3">ex-ante</td></tr>
<tr><td>Analysis Technique</td><td colspan="2">expert-based</td><td colspan="2">rule-based</td><td colspan="2">indicator-based</td></tr>
<tr><td>Analysis Concern</td><td colspan="3">functional</td><td colspan="3">non-functional</td></tr>
<tr><td>Self-Referentiality</td><td colspan="2">none</td><td colspan="2">single-level</td><td colspan="2">multi-level</td></tr>
</table>

Table 1: Classification schema for EA analysis approaches

Body of Analysis: **structure**, **behavior statistics**, and **dynamic behavior**

As discussed in Section 1, enterprises are complex systems, which encompass a dense web of interconnections. Together with the sheer number of constituents, this number of interconnections contributes to the *structural complexity* of the enterprise system, reflected in the corresponding EA models. EA analyses have to deal with this complexity by e.g. providing mechanisms to aggregate concepts to a higher level of abstraction or to overleap concepts in transitive relationships. Aside the complex structure, the enterprise system's constituents exert a *complex behavior*, which also forms a valuable object for analyses. In particular, statistic information about the system's behavior may be highly relevant to the stakeholders of EA management. Such analyses present behavioral aspects in an aggregated manner, e.g. as *moving-average* of a certain behavioral attribute. This statistic information provides an aggregation of lower level information concerned with the dynamic behavior of the system's constituents, i.e. the time-series of values of an attribute or a stream of discrete events. The *dynamic behavior* is especially an interesting object of analysis, if *pathological* effects in the behavior are considered, e.g. the impact of a system failure propagating over time.

Time Reference: **ex-post** and **ex-ante**
EA analyses are performed on models of the architecture. Regarding these architectural models, two types can be distinguished: models of *current architectures* and of *planned architectures*. Models of the first type describe the architecture of an existing, i.e. implemented EA, while models of the second type mainly refer to planning scenarios for the architecture. In this respect, analyses have to act differently on the two types of models – in the case with current architectures, both information on the structure and the behavior is available, i.e. can be measured[2]. In contrast, architectural plans describe the planned structure, where information on the resulting behavior is not present, as the architecture scenario is neither implemented nor operated. But not only for the question of behavior the distinction between *ex-post* analyses, i.e. ones targeting current architectures, and *ex-ante* analyses on planned architectures is sensible. The same is true for analyses on the structure, which is fixed for current architectures, while architectural scenarios are inevitably connected to a higher level of uncertainty. They hence have at least a partially predictive character. The aforementioned facts motivate the distinction concerning the point in time(*ex-post* and *ex-ante*).

Analysis Technique: **expert-based**, **rule-based**, and **indicator-based**
EA analyses can employ different levels of formalization ranging from informal expert-based techniques to formal indicator-based ones. The *expert-based* approaches are the most flexible but also most time-consuming ones, and depend on the experience and expertise of the executing person. Therein, one or more EA experts analyze properties of the EA along appropriate architecture views, as e.g. reports or graphical visualizations. Such analyses may produce results that range from rather concrete advices to more general and abstract ideas for future architecture development. *Rule-based* analyses are performed at an increased level of formalization and can consistently be automated. Thereby, rules describe certain architectural constellations, which are either desirable or should be avoided, i.e. they can represent *patterns* or *anti-patterns* for an EA. An even higher level of formalization is reached by *indicator-based* analysis techniques. Where rule-based techniques can only assess the absence or presence of certain architectural patterns, indicators can be used to quantitatively assess architectural properties, as e.g. complexity. Commonly, the values of these indicators are derived from the values of observable architecture properties by computation. While producing the most expressive and most directly interpretable analysis results, indicator-based techniques are narrow in their focus and their results have to be interpreted carefully. In particular, the analysts must keep in mind,

[2] Nevertheless, the information might not be collected due to complexity or cost reasons.

that an indicator is always based on assumptions on the architecture, who – if not longer valid – deprive the indicator of its expressive power.

Analysis Concern: functional and **non-functional**
Enterprises are designed towards performing enterprise functions, such as production or sales. Thereby, we can assume that an EA is developed with certain *functional* requirements in mind. With the help of EA analysis, the fulfillment of these requirements by the enterprise system, expressed in its architecture, can be assessed. In contrast, the enterprise system also has *non-functional* properties, as e.g. execution times or throughput. These properties might also be interesting for analysis, especially in cases, where the fulfillment of the functional requirements is given and alternatives for achieving this fulfillment should be compared. Some EA management practitioners mention *economic concerns* [11], as e.g. operating or maintenance costs, and want those to be analyzed especially for EA planning. We nevertheless do not further distinguish between non-functional and economic concerns, as we regard the latter to be a specialization of the former. The distinction between functional and non-functional concerns resembles the similar distinction as discussed in the discipline of software engineering, where also different analysis approaches exist targeting both types of aspects (cf. e.g. [20]). For the context of the EA, this distinction is also undertaken in [21], where *functional* and *quantitative* analyses are juxtaposed. We argue that the terms functional and non-functional are more appropriate, as *quantative* alludes to a certain type of analysis result, but does not make prescriptions on the analyzed concern. Put in other words, a non-functional requirement as e.g. security could be analyzed in a non-quantative way.

Consideration of Self-Referentiality: **None**, **Single-level**, and **Multi-level**
As shown in Figure 1, organizational structures as well as strategic planning concepts are also considered to be part of the EA. In this vein, the element of *self-referentiality* is introduced into the field of EA management in general and EA analysis in special. In particular, the groups and roles in the enterprise actually performing the EA analysis can also be modeled in the corresponding EA model. The situation can become even more complex, as the activities for managing, describing, and planning the EA can be managed, described, and planned as parts of an EA. This agrees with the interpretation of the enterprise as a *living system* having the property of *autopoiesis*. Such system is in accordance to [22] defined as:

> a [system] organized as a network of processes of production, transformation and destruction of components which:
> (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them, and
> (ii) constitute it [the system] as a concrete unity in space in which they (the components) exist by specifying the topological domain of its realization as such a network.

Regarding this self-creating and -updating characteristics of the EA, an approach to EA analysis has to decide, to which extent these higher-level interactions are considered. In a simple model, the architectural aspects behind the activities of EA management are not considered, i.e. there is no consideration of self-referentiality. A more elaborate analysis model would include one level of self-referentiality, i.e. it would analyze the EA management activities as EA constituents. A multi-level approach goes even further and considers the activities of the meta-processes of EA management (such as *EA management governance*, cf. [23]) to be part of the EA model. Introducing EA related activities into the EA itself is very likely to increase the complexity of the corresponding analyses, but may simplify the understanding of *emergent* behavior [24] in the enterprise system.

In [21] another *analysis dimension* is presented, distinguishing between *analytical* and *simulation* analysis techniques. Thereby, simulation-based analysis techniques are understood as *executing a model* for performing the analysis, while analytical techniques aim at finding a closed solution for an analysis model. In contrast, we do not estimate the importance of that distinction high enough to be beneficial for our subsequent classification.

3 State-of-the-art in enterprise architecture analysis

Based on the classification schema introduced in preceding Section 2, in the following the state-of-the-art in EA analysis is investigated. In particular, prominent approaches to this field are introduced and revisited in respect to the classification dimensions of the schema.

In [25], an approach to perform EA analysis with XML is presented. The article hence puts a strong emphasis on issue how to represent the EA in a model for performing analyses. This model is rather fine grained, proposing to use *state-machines* to include behavioral descriptions as well as XML elements to represent structural information about the architecture. The time reference in the analyses is not explicitly alluded to, although the structural analyses can be applied both ex-ante and ex-post, while the state-machine based modeling of EA behavior points towards an ex-ante analysis of behavioral aspects. The question how to calibrate behavior models against measured behavior is nevertheless not discussed in the article. Also, the question of the employed analysis technique is not detailed, but indications pointing towards rule-based analysis techniques exist. When it comes to the possible analysis concerns, the paper does neither advocate functional nor non-functional properties of the EA. The approach presented nevertheless can be regarded generic enough to support both types of concerns. Self-referentiality of EA analysis and management is not discussed in the article. This leads to a classification of the approach as shown in Table 2.

A more quantitative approach to EA analysis is proposed in [26], where exemplary indicators for analyzing EA behavior are provided. These indicators are developed to provide aggregated information on the behavior of certain EA artifacts, i.e. to compute behavior statistics. The analysis model presented in the ap-

<table>
<tr><td>Body of Analysis</td><td>structure</td><td colspan="2"></td><td>dynamic behavior</td></tr>
<tr><td>Time Reference</td><td colspan="2">ex-post</td><td colspan="2">(ex-ante)</td></tr>
<tr><td>Analysis Technique</td><td></td><td colspan="2">rule-based</td><td></td></tr>
<tr><td>Analysis Concern</td><td colspan="2">(functional)</td><td colspan="2">(non-functional)</td></tr>
<tr><td>Self-Referentiality</td><td>none</td><td colspan="2"></td><td></td></tr>
</table>

Table 2: Classification of EA analysis approach of [25]

proach integrates into the ArchiMate architecture description language[3], whose concepts are augmented with additional properties reflecting non-functional aspects of the EA, as e.g. the execution or completion time for a business process. The time reference in the analysis is discussed alongside the question of the quantitative input for the analysis model. In particular, the authors highlight that measuring the behavior of an existing system can provide valuable input, but also rise the question of reproducible circumstances for the measurements. For systems, which are still to be developed, estimates for properties, e.g. based on comparable architectures, are noted as possible source for quantitative information. Based on this information, performance measures for the (planned) system are derived analytically, i.e. the values for descriptive indicators are computed. The question of self-referentiality of EA analysis and management is not alluded to in the article, whose approach can be classified as shown in Table 3.

<table>
<tr><td>Body of Analysis</td><td></td><td colspan="2">behavior statistics</td><td></td></tr>
<tr><td>Time Reference</td><td colspan="2">ex-post</td><td colspan="2">ex-ante</td></tr>
<tr><td>Analysis Technique</td><td></td><td colspan="2"></td><td>indicator-based</td></tr>
<tr><td>Analysis Concern</td><td colspan="2"></td><td colspan="2">non-functional</td></tr>
<tr><td>Self-Referentiality</td><td>none</td><td colspan="2"></td><td></td></tr>
</table>

Table 3: Classification of EA analysis approach of [26]

Two prominent approaches for EA analysis are presented in [27] and [28], respectively. These approaches vary significantly concerning their origins and concerning the concepts, which they employ. Concerning the classification schema, both approaches are nevertheless quite similar and fit into one classification as shown in Table 4. The two approaches support both ex-ante and ex-post analyses of enterprise systems represented in EA models. In particular, [27] emphasizes the importance of an indicator system, which is used for analyses there, as means to support communication in an enterprise. The focus of both approaches lies on statistic information arising from the behavior of the enterprise system, although to a limited amount also structural aspects of the EA are analyzed. Beside the indicator-based analysis technique, a prominent difference exists: the approach of [28] aims at the development of single indicators, while [27] goes even further. In particular, the latter approach seeks to develop integrated and consistent in-

[3] For more information see http://archimate.telin.nl.

dicator systems. Regarding the analysis concern, both approaches focus on non-functional requirements, with the approach of [27] centering on more economic indicators, whereas [28] puts an emphasis on *classical* non-functional aspects, such as availability. Finally, it can be said that the approaches restrict themselves to analyses of the EA; the self-referential character of EA management is hence not considered.

<table>
<tr><td>Body of Analysis</td><td colspan="2">(structure)</td><td colspan="2">behavior statistics</td><td colspan="2"></td></tr>
<tr><td>Time Reference</td><td colspan="3">ex-post</td><td colspan="3">ex-ante</td></tr>
<tr><td>Analysis Technique</td><td colspan="2"></td><td colspan="2"></td><td colspan="2">indicator-based</td></tr>
<tr><td>Analysis Concern</td><td colspan="3"></td><td colspan="3">non-functional</td></tr>
<tr><td>Self-Referentiality</td><td colspan="2">none</td><td colspan="2"></td><td colspan="2"></td></tr>
</table>

Table 4: Classification of EA analysis approaches of [27] and [28]

The topic of EA analysis is also discussed in [11]. There an emphasis is put on the utilization of EA models, which have been created during other EA management activities. These models reflect structural aspects of the EA, thus limiting the body of analysis on such aspects. Furthermore, existing enterprise systems reflected in their EAs are analyzed according to different concerns. The majority of these concerns is functional, e.g. homogeneity of the application landscape or the interdependencies between the business applications are considered. Complementing, two economic concerns, namely *costs* and *benefits*, are alluded to as typical EA analysis concerns. Nevertheless, the measures for those costs and benefits remain on a rather abstract level. The different analysis concerns are addressed by different techniques, of which the majority is expert-based, i.e. they utilize specific viewpoints to present architecture information to an enterprise architect, who informally assesses e.g. the level of homogeneity. When it comes to the economic concerns, the predominance of the expert-based analysis is broken in favor of a few quantitative indicators, which are especially used to operationalize benefits. All the different types of analysis as proposed in [11] target the enterprise system as reflected in its EA – the corresponding management system is not discussed in the analyses. This leads to a classification as shown in Table 5.

<table>
<tr><td>Body of Analysis</td><td colspan="2">structure</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>Time Reference</td><td colspan="3">ex-post</td><td colspan="3"></td></tr>
<tr><td>Analysis Technique</td><td colspan="2">expert-based</td><td colspan="2"></td><td colspan="2">(indicator-based)</td></tr>
<tr><td>Analysis Concern</td><td colspan="3">functional</td><td colspan="3">(non-functional)</td></tr>
<tr><td>Self-Referentiality</td><td colspan="2">none</td><td colspan="2"></td><td colspan="2"></td></tr>
</table>

Table 5: Classification of EA analysis approach of [11]

Agreeing with the overall understanding of EA analysis as proposed in [11] and discussed in [29], a dedicated support of *impact analyses* is shown in [30]. In particular, the EA is thereby understood as a directed graph reflecting the structure of the enterprise system. On this graph, rule-based analyses are performed to assess and evaluate the transitive impact of an EA constituent, e.g. in cases of failure. Thereby, the analyses are applied to current architectures, although the proposed method is not limited in this respect and could hence also be applied on planned architectures. Regarding the analysis concerns, the approach does not make assumptions, i.e. it can handle both functional and non-functional ones. The following two analysis questions taken from [30] exemplify this: *Which business applications are used during the creation of a selected product?* and *Which applications fail, if a certain server fails?*. Due to its high generality, the approach would not be restricted to pure EA analyses, but could also employ higher level processes and concepts. Nevertheless, the paper does not account for this method but limits the approach to structural analyses of the enterprise system. This leads to a classification as shown in Table 6.

<table>
<tr><td>Body of Analysis</td><td>structure</td><td colspan="2"></td><td></td></tr>
<tr><td>Time Reference</td><td colspan="2">ex-post</td><td colspan="2">(ex-ante)</td></tr>
<tr><td>Analysis Technique</td><td></td><td colspan="2">rule-based</td><td></td></tr>
<tr><td>Analysis Concern</td><td colspan="2">functional</td><td colspan="2">non-functional</td></tr>
<tr><td>Self-Referentiality</td><td>none</td><td colspan="2"></td><td></td></tr>
</table>

Table 6: Classification of EA analysis approach of [30]

Quite different from the aforementioned analysis approaches is the approach presented in [31], which centers around the idea of *intentional modeling*. Based on this idea, not only the EA as the architecture of the enterprise system, but also the architecture behind the EA management processes is analyzed. This especially applies to the process of constructing an EA model, whose motivation and stakeholders are heavily considered. To foster the analyses, intentional modeling concepts from the *i* language* (cf. [32]) are linked to structural concepts of the EA, such as (business) processes. The models used therein are models of future EAs (and their related EA management processes), which are developed from current architecture models that were annotated with intentional meta-information. The actual analyses are performed by EA experts, as they can account for the different kinds of goals from the intentional models. Especially the so called *soft-goals* demand expert analysis, as they are not easily operationalized. So expert interviews are strongly alluded to as a suitable analysis technique. In most cases, functional properties of the enterprise system are considered during the analyses, although viability analysis (cf. e.g. [33]) further allows to include non-functional factors, as performance or security into the analysis. The classification of the approach according to the schema from Section 2 reads as shown in Table 7.

<table>
<tr><td>Body of Analysis</td><td colspan="2">structure</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>Time Reference</td><td colspan="3"></td><td colspan="3">ex-ante</td></tr>
<tr><td>Analysis Technique</td><td colspan="2">expert-based</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>Analysis Concern</td><td colspan="3">functional</td><td colspan="3">(non-functional)</td></tr>
<tr><td>Self-Referentiality</td><td colspan="2"></td><td colspan="2">single-level</td><td colspan="2"></td></tr>
</table>

Table 7: Classification of EA analysis approach of [31]

Summarizing the above classifications of the selected EA analysis approaches, we can show a classification schema referencing the corresponding approaches for the different characteristics in the dimensions of classification. This schema is shown in table 8 to prepare the subsequent discussions on the state-of-the-art in EA analysis in the subsequent section.

<table>
<tr><td>Body of Analysis</td><td colspan="2">structure
[11],[25],([27],[28])
[30],[31]</td><td colspan="2">behavior statistics
[26],[27],[28]</td><td colspan="2">dynamic behavior
[25]</td></tr>
<tr><td>Time Reference</td><td colspan="3">ex-post
[11],[25],[26],[27],[28],[30]</td><td colspan="3">ex-ante
([25]),[26],[27],[28],([30]),[31]</td></tr>
<tr><td>Analysis Technique</td><td colspan="2">expert-based
[11],[31]</td><td colspan="2">rule-based
[25],[30]</td><td colspan="2">indicator-based
([11]),[26],[27],[28]</td></tr>
<tr><td>Analysis Concern</td><td colspan="3">functional
[11],([25]),[30],[31]</td><td colspan="3">non-functional
([11],[25]),[26],[27],
[28],[30],([31])</td></tr>
<tr><td>Self-Referentiality</td><td colspan="2">none
[11],[25],[26],[27],
[28],[30]</td><td colspan="2">single-level model
[31]</td><td colspan="2">multi-level model</td></tr>
</table>

Table 8: Summarizing classification of the EA analysis approaches

4 Conclusion and Outlook

In this paper, we presented a classification schema for EA analysis approaches and applied this schema to the state-of-the-art in the field. Thereby, we could show that the different characteristics for almost every dimension of classification are incorporated into at least one approach. The sole exception to this finding is *multi-level* self-referentiality, which actually is not easy to implement, especially as EA management is a relatively new discipline without well-established meta-processes (cf. also [23]). Aside this apparent lack of a supporting approach for meta-EA management analyses, an interesting property can be discovered by considering the overall coverage reached by the different approaches. Subsequently, we discuss this on the different dimensions of classification:

Body of Analysis: Predominantly, structural aspects of the EA such as interconnections, are analyzed. This might indebt to the fact, that behavioral information is less frequently collected and mostly used in an aggregated form, i.e. as behavior statistics. Analyzing the dynamics in behavior, e.g. pathological effects, is only discussed in the approach of [25], although the description of the actual analysis method remains vague there. Furthermore, dynamic behavior and behavior statistics are not considered in combination by one of the approaches. We nevertheless see that these bodies of analysis are intrinsically linked to each other.

Time Reference: Many approaches analyzed are not specifically designed towards ex-ante or ex-post analyses only, but can be applied in both time references equally. This is not surprising, as most of the approaches focus on structural aspects, which can be handled irrespective of their planning character. When the question comes to approaches analyzing aggregated enterprise behavior, the situation presents itself a bit more complicated. Most likely, the corresponding approaches support ex-post and ex-ante analyses, but refer to the latter only in a side-note, reading like the following: *for applying the technique on planned architectures, estimations have to be made.* The question, how architectural properties for future EAs can be estimated is hence not alluded to in detail in the majority of of the approaches.

Analysis Technique: The three characteristics of this dimension are well-applied in the different analysis approaches. On the one hand, this might be caused by the fact that a clear distinction between expert-based analyses and the other characteristics is not always possible based on the approaches' descriptions. When it comes to rule-based and indicator-based, the situation is somewhat different. The approaches are either rule- or indicator-based, but none of them combines both analysis techniques.

Analysis Concern: Non-functional and especially economic aspects of the enterprise system are clearly more in the focus of EA analysis approaches, although especially some highly generic approaches are applicable on both types of concerns. Nevertheless, these approaches are most likely to apply the expert-based analysis technique. This might ascribe to the fact that describing the function of an EA constituent is not that easy to perform and hence more structured analysis techniques are hard to apply.

Self-Referentiality: Only one of the considered approaches goes beyond a pure EA analysis by also considering the complementing EA management process. This might have manifold reasons aside the aforementioned novelty of the discipline. To name just another important reason, it has to be noted that neither a commonly accepted definition of the EA management function yet exists nor a reference process, one could evaluate the EA management against. This relates the question of self-referentiality in EA analysis approaches to the new and emerging field of EA maturity models (see e.g. [34]), which is under heavy development for a couple of years now.

Concluding it can be said, that the field of EA analysis shows all signs of an emerging discipline. Quite a few approaches to the field exist, differing in origin, technique, and coverage. But as of today, these approaches are not well-linked to each other, as they e.g. employ many different modeling techniques. Advancing the field of EA analysis would hence clearly mean, bringing together the different approaches especially concerning the dimensions *body of analysis*, *time reference*, and *analysis technique*. In particular, the first two dimensions are closely related in this respect – linking dynamic behavior models with behavior statistics models could advance the development of ex-ante analysis. The dynamic models would allow simulation of behavior of planned architectures, thus replacing the vague guessing of estimates, which is widely alluded to. From these dynamic models, behavior statistics could be derived analytically or by simulation. Thereby, the experienced gap in these dimensions could be closed.

In respect to the analysis techniques, a similar argument applies. Certain architecture properties, most likely structural ones, can be easily assessed by rule-based techniques. With the help of these techniques architectural anti-patterns could be matched, ruling out certain planned architectures in advance. After this, more detailed indicator-based analysis techniques could be applied to predict non-functional properties of the planned EA. Thereby, an integrated analysis approach would be provided, allowing to narrow the *search space* by ruling out some options before potentially applying extensive and time-consuming simulation techniques.

Above discussions outline a line of action for future research, in which the somewhat isolated EA analysis approaches are integrated into a conceptual framework. We regard the classification schema as presented in Section 2 to give quite a few indications on how such a framework could be structured. Nevertheless, some challenges are await on the way towards an integrated analysis approach. In particular, we expect the question of a conceptual modeling language for supporting arbitrary EA analyses an interesting one. This would especially be true, if such language was designed to not burden the related EA management activities, as EA documentation or EA planning, with formal mechanisms that are solely needed for EA analyses.

References

1. Wittenburg, A.: Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften. PhD thesis, Fakultät für Informatik, Technische Universität München (2007)
2. Wittenburg, A., Matthes, F., Fischer, F., Hallermeier, T.: Building an integrated it governance platform at the bmw group. International Journal Business Process Integration and Management **2**(4) (2007)
3. Matthes, F., Buckl, S., Leitel, J., Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008. Chair for Informatics 19 (sebis), Technische Universität München, Munich (2008)
4. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2008)

5. Fischer, R., Aier, S., Winter, R.: A federated approach to enterprise architecture model maintenance. In: Enterprise Modelling and Information Systems Architectures - Concepts and Applications, Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), St. Goar, Germany, October 8-9, 2007. (2007) 9–22
6. Frank, U.: Multi-perspective enterprise modeling (memo) – conceptual framework and modeling languages. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 3003. (2002) 1258–1267
7. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Heidelberg, Germany (2005)
8. Dern, G.: Management von IT-Architekturen (Edition CIO). Vieweg, Wiesbaden (2006)
9. Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.P.: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten. dpunkt.verlag, Heidelberg (2008)
10. Keller, W.: IT-Unternehmensarchitektur. dpunkt.verlag, Heidelberg (2007)
11. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner, Wiesbaden, Germany (2006)
12. van der Torre, L.W.N., Lankhorst, M.M., ter Doest, H.W.L., Campschroer, J.T.P., Arbab, F.: Landscape maps for enterprise architectures. In Dubois, E., Pohl, K., eds.: Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006, Luxembourg, Luxembourg, June 5-9, 2006, Proceedings. Volume 4001 of Lecture Notes in Computer Science., Springer (2006) 351–366
13. Department of Defense (DoD) USA: DoD Architecture Framework Version 1.5: Volume I: Definitions and Guidelines. `http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf` (cited 2009-06-30) (2007)
14. Murer, S., Worms, C., Furrer, F.: Managed evolution. Informatik Spektrum **31**(6) (2008) 527–536
15. Harmon, K.: The ”systems” nature of enterprise architecture. In: IEEE International Conference on Systems, Man and Cybernetics 2005. (2005) 78–58
16. Forrest, J.W.: Industrial Dynamics. MIT Press (1961)
17. Pollak, B.: Ultra-Large-Scale Systems – The Software Challenge of the Future. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, USA (2006)
18. Jansen-Vullers, M., Netjes, M.: Business process simulation – a tool survey. In: Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Denmark (2006)
19. Laguna, M., Marklund, J.: Business Process Modeling, Simulation, and Design. Prentice Hall PTR, Indianapolis (2004)
20. Sommerville, I.: Software Engineering. 6 edn. Pearson Studium, Munich (2001)
21. Lankhorst, M.: Introduction to enterprise architecture. In: Enterprise Architecture at Work, Berlin, Heidelberg, New York, Springer (2005)
22. Varela, F.G., Maturana, H.R., Uribe, R.: Autopoiesis: the organization of living systems, its characterization and a model. Currents in modern biology **5**(4) (May 1974) 187–196
23. Buckl, S., Schweda, C.: A viable system perspective on enterprise architecture management. In: 2009 IEEE International Conference on Systems, Man, and Cybernetics. (2009) (in publication).
24. Baas, N.A., Emmeche, C.: On emergence and explanation. Working Papers 97-02-008, Santa Fe Institute (1997)

25. de Boer, F.S., Bonsangue, M.M., Jacob, J., Stam, A., van der Torre, L.: Enterprise architecture analysis with xml. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005). Volume 8., Los Alamitos, CA, USA, IEEE Computer Society Press (2005) 222b
26. Iacob, M.E., Jonkers, H.: Quantitative analysis of enterprise architectures. In Konstantas, D., Bourrières, J.P., Léonard, M., Boudjlida, N., eds.: Interoperability of Enterprise Software and Applications, Geneva, Switzerland, Springer (2006) 239–252
27. Frank, U., Heise, D., Kattenstroth, H., Schauer, H.: Designing and utilising business indicator systems within enterprise models – outline of a method. In: Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management 27.-28. November 2008, Saarbrücken, Germany (2008)
28. Johnson, P., Nordström, L., Lagerström, R.: Formalizing analysis of enterprise architecture. In: Interoperability for Enterprise Software and Applications Conference, Bordeaux, France, Springer (2006) 10
29. Bucher, T., Fischer, R., Kurpjuweit, S., Winter, R.: Analysis and application scenarios of enterprise architecture: An exploratory study. In: Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006), 16-20 October 2006, Hong Kong, China, Workshops, Washington, DC, USA, IEEE Computer Society (2006)
30. Kurpjuweit, S., Aier, S.: Ein allgemeiner Ansatz zur Ableitung von Abhängigkeitsanalysen auf Unternehmensarchitekturmodellen. In: 9. Internationale Tagung Wirtschaftsinformatik (WI 2007), Wien, Österreichische Computer Gesellschaft (2009) 129–138
31. Yu, E., Strohmaier, M., Deng, X.: Exploring intentional modeling and analysis for enterprise architecture. In: Proceedings of the EDOC 2006 Conference Workshop on Trends in Enterprise Architecture Research (TEAR 2006), Hong Kong, IEEE Computer Society Press (2006) 32–32
32. Yu, E.: Modelling strategic relationships for process reengineering. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada (1996)
33. Yu, E.: Towards modeling and reasoning support for early-phase requirements engineering. In: RE ’97: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Washington, DC, USA, IEEE Computer Society (1997)
34. van Steenbergen, M., van den Berg, M., Brinkkemper, S.: An instrument for the development of the enterprise architecture practice. In Cardoso, J., Cordeiro, J., Filipe, J., eds.: ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume EIS, Funchal, Madeira, Portugal, June 12-16, 2007 (3). (2007) 14–22